

FileInASnap MiniMax - AI-Powered File Management

Deploy Active

AI Powered

MCP Server

A modern, AI-enhanced file management application inspired by SparkleShare, featuring intelligent document analysis, smart categorization, and automated organization powered by OpenRouter's free AI models.

Live Demo




 [Try FileInASnap MiniMax](#)






Features

Core File Management

- **Smart Folders** - Intelligent file organization with AI-powered suggestions
- **Real-time Sync** - Instant file updates and synchronization
- **Multi-language Support** - Available in 6 languages (EN, ES, FR, DE, ZH, JA)
- **SparkleShare-inspired UI** - Clean, modern interface with intuitive navigation

AI-Powered Capabilities

-  **Document Analysis** - Extract entities, categorize content, and identify document types
-  **Smart Tagging** - Automated tag generation based on content analysis
-  **File Categorization** - Intelligent classification using content-aware AI models

-  **Image Analysis** - Computer vision with object detection and OCR capabilities
-  **Duplicate Detection** - Advanced similarity detection beyond simple checksums
-  **Content Summarization** - AI-generated summaries for any file type
-  **Organization Suggestions** - AI-recommended folder structures and workflows
-  **Real-time Monitoring** - Live status of all AI system capabilities

Architecture

Frontend (React + Vite + TypeScript)

- **Framework:** React 18 with TypeScript
- **Build Tool:** Vite for fast development and optimized builds
- **Styling:** TailwindCSS with custom SparkleShare-inspired theme
- **UI Components:** Custom component library with accessibility features
- **Internationalization:** Multi-language support with dynamic switching

Backend (Supabase + Edge Functions)

- **Database:** PostgreSQL with real-time subscriptions
- **Authentication:** Supabase Auth with social providers
- **Storage:** Secure file storage with access controls
- **API:** Serverless Edge Functions for AI integration

AI Integration (MCP Server)

- **Primary Provider:** OpenRouter free tier models
- **Fallback Ready:** MiniMax integration scaffolded for future activation
- **Models:** Mistral-7B, Llama-3.1-8B, Qwen2-7B for text processing

- **Architecture:** Universal provider abstraction for easy switching

Quick Start

Prerequisites

- Node.js 18+
- npm or pnpm
- Supabase account
- OpenRouter API key (free tier available)

Installation

1. Clone the repository

```
bash git clone https://github.com/Greenmamba29/
fileinasnap_minimax.git cd fileinasnap_minimax
```

2. Install dependencies

```
bash npm install # or pnpm install
```

3. Environment Setup

Copy `.env.example` to `.env.local` and configure:

```
env VITE_SUPABASE_URL=your_supabase_url
VITE_SUPABASE_ANON_KEY=your_supabase_anon_key
OPENROUTER_API_KEY=your_openrouter_api_key
```

4. Start Development Server

```
bash npm run dev # or pnpm dev
```

5. Build for Production

```
bash npm run build # or pnpm build
```

MCP Server Setup

The AI capabilities are powered by a custom MCP (Model Context Protocol) server located in the `./mcp-server/` directory.

Local MCP Server Development

1. Navigate to MCP server directory

```
bash cd mcp-server
```

2. Install Python dependencies

```
bash pip install -r requirements.txt
```

3. Configure environment

```
bash export OPENROUTER_API_KEY="your_openrouter_api_key"
```

4. Run MCP server

```
bash python server.py
```

Available AI Tools

- `analyze_document` - Document analysis with entity extraction
- `categorize_files` - Batch file classification
- `generate_file_tags` - Intelligent tagging system
- `analyze_image` - Computer vision analysis
- `detect_file_duplicates` - Advanced duplicate detection
- `generate_file_summary` - Content summarization
- `suggest_folder_structure` - Organization suggestions
- `get_provider_status` - System health monitoring

UI/UX Design

SparkleShare Inspiration

- **Color Palette:** Clean whites, soft grays, and accent blues
- **Typography:** Modern, readable fonts with proper hierarchy
- **Icons:** Custom icon set resembling SparkleShare's folder metaphors
- **Layout:** Grid-based design with intuitive navigation patterns

AI Integration Indicators

- **Sparkle Icons** - Subtle indicators for AI-powered features
- **Loading States** - Progress indicators for AI operations
- **Result Displays** - Clean presentation of AI insights and suggestions

Development

Project Structure

```
fileinasnap_minimax/
├── src/                # Frontend source code
│   ├── components/    # React components
│   ├── pages/         # Page components
│   ├── hooks/         # Custom React hooks
│   ├── utils/         # Utility functions
│   └── styles/        # CSS and styling
├── public/            # Static assets
├── supabase/          # Supabase configuration
│   ├── functions/     # Edge Functions
│   └── migrations/    # Database migrations
├── mcp-server/        # AI MCP Server
│   ├── src/           # Python source code
│   ├── examples/      # Usage examples
│   └── tests/         # Test suite
└── locales/           # Internationalization files
```

Key Technologies

- **Frontend:** React, TypeScript, Vite, TailwindCSS
- **Backend:** Supabase, PostgreSQL, Edge Functions
- **AI:** OpenRouter APIs, MCP Protocol
- **Deployment:** Vercel/Netlify (Frontend), Supabase (Backend)

Deployment

The application is deployed using modern cloud platforms:

- **Frontend:** Automatically deployed from main branch

- **Backend:** Supabase Edge Functions for serverless scaling
- **AI Services:** OpenRouter free tier with MiniMax upgrade path

Contributing

We welcome contributions! Please see our contributing guidelines:

1. Fork the repository
2. Create a feature branch (`git checkout -b feature/amazing-feature`)
3. Commit your changes (`git commit -m 'Add amazing feature'`)
4. Push to the branch (`git push origin feature/amazing-feature`)
5. Open a Pull Request

License

This project is licensed under the MIT License - see the [LICENSE](#) file for details.

Acknowledgments

- **SparkleShare** - UI/UX design inspiration
- **OpenRouter** - Free AI model access
- **MiniMax** - Future AI capabilities integration
- **Supabase** - Backend infrastructure
- **React Team** - Amazing frontend framework

Support

- **Issues:** [GitHub Issues](#)
 - **Discussions:** [GitHub Discussions](#)
 - **Email:** support@fileinasnap.com
-

